Data Mining & Knowledge Discovery

Lesson 7 Classification (I)

Lan Man
Department of Computer Science and Technology
East China Normal University

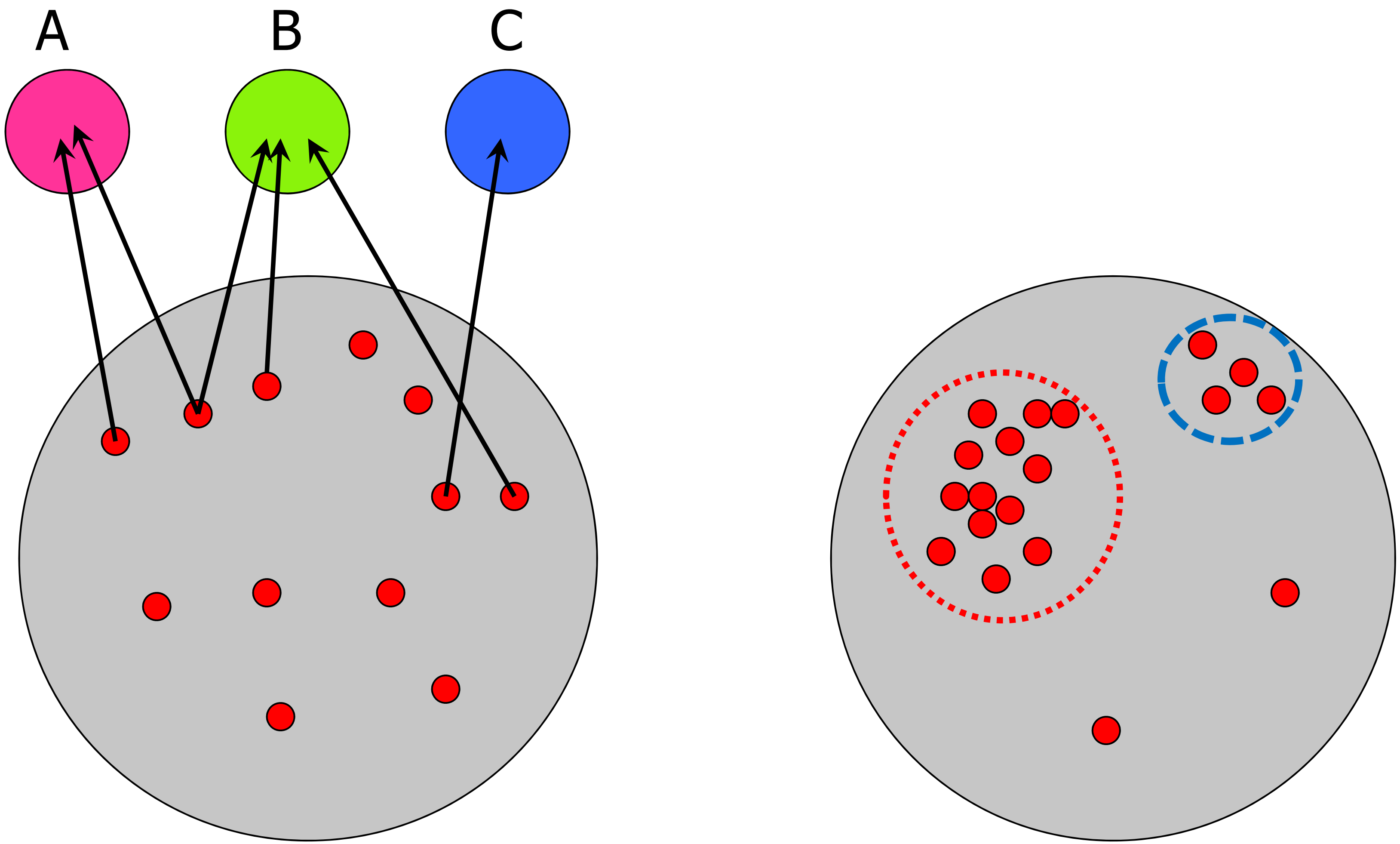
Lesson 7 Classification (I)

- Basic Concepts
- Decision Tree Induction
- Bayesian Classification
- Backpropagation
- Support Vector Machines (SVM)
- Lazy Learners (kNN)
- Other Classification Methods
- Additional Topics
- Prediction
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Summary

Supervised vs. Unsupervised Learning

- **Supervised learning (classification)**
  - Supervision: The training data (observations, measurements, etc.) are accompanied by **class labels** indicating the class of the observations
  - New data is classified based on the training set
- **Unsupervised learning (clustering)**
  - The class labels of training data is **unknown**
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

Classification vs Clustering



Classification vs. Numeric Prediction

- Classification
  - predicts categorical class labels (discrete or nominal)
  - constructs a model based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data
- Prediction
  - models continuous-valued functions, i.e., predicts unknown or missing values
- Typical applications
  - Credit approval, Target marketing, Medical diagnosis, Fraud detection, Performance prediction

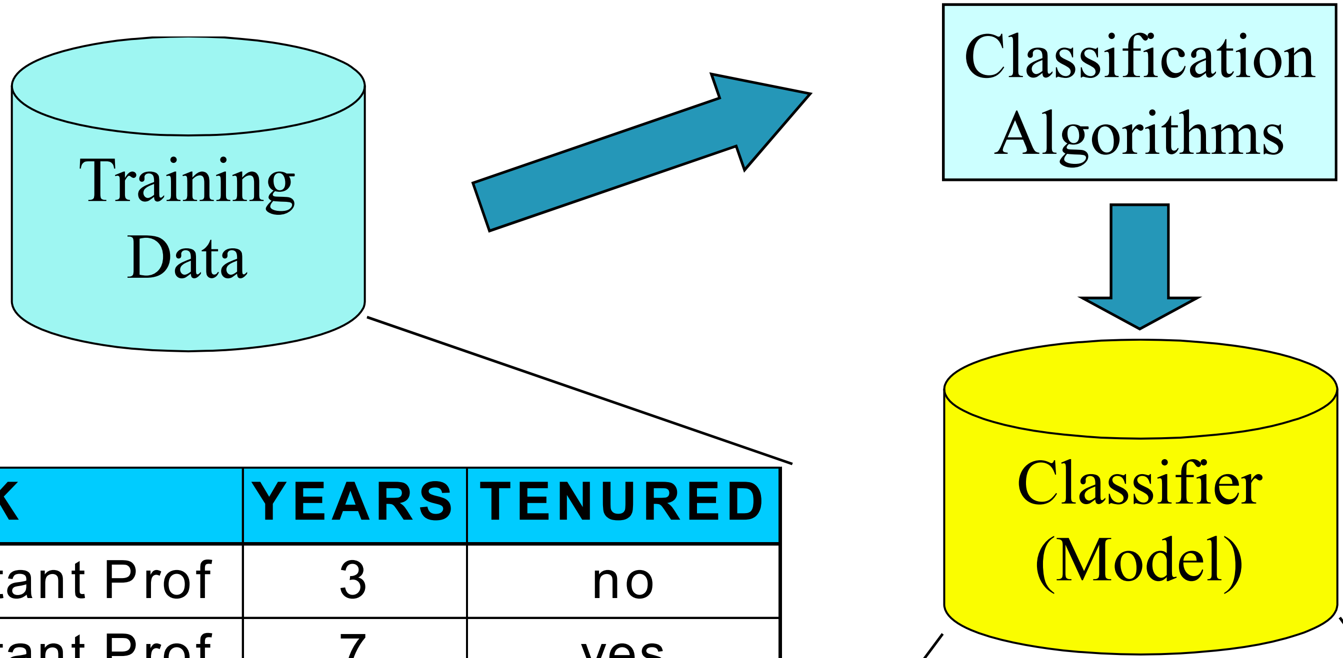
Classification—A Two-Step Process

- **Step 1. Model construction**: describing a set of predetermined classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
  - The set of tuples used for model construction is training set
  - Each tuple is one training sample
  - The model is represented as classification rules, decision trees, or mathematical formulae

Classification—A Two-Step Process

- **Step 2. Model usage**: for classifying future or unknown objects
  - Estimate accuracy of the model
    - The known label of test sample is compared with the classified result from the model
    - Accuracy is the percentage of test set samples that are correctly classified by the model
    - Test set is independent of training set (otherwise overfitting)
  - If the accuracy is acceptable, use the model to classify new data whose class labels are not known
- Note: If *the test set* is used to select models, it is called validation (test) set

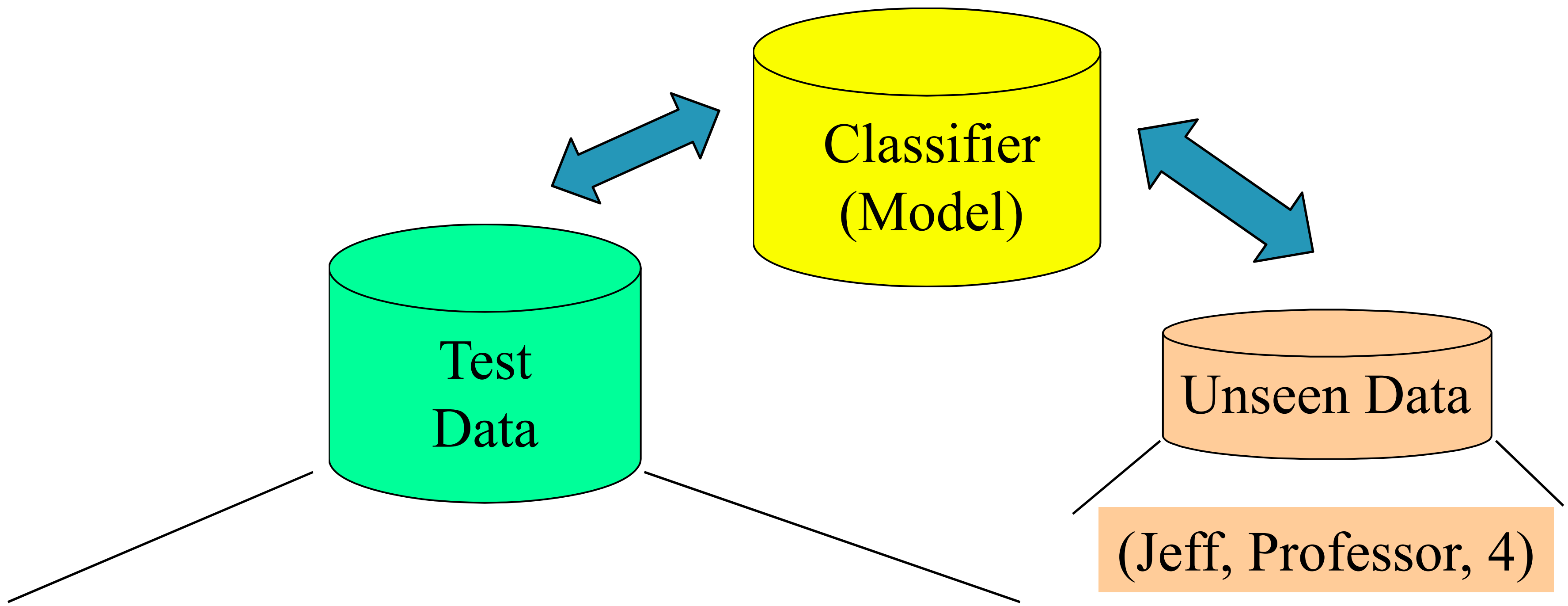
Process (1): Model Construction



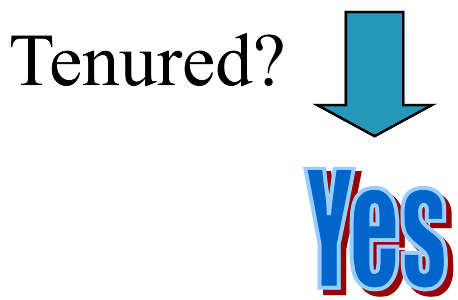
NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’

Process (2): Using the Model in Prediction



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes



Example

2. Evaluation

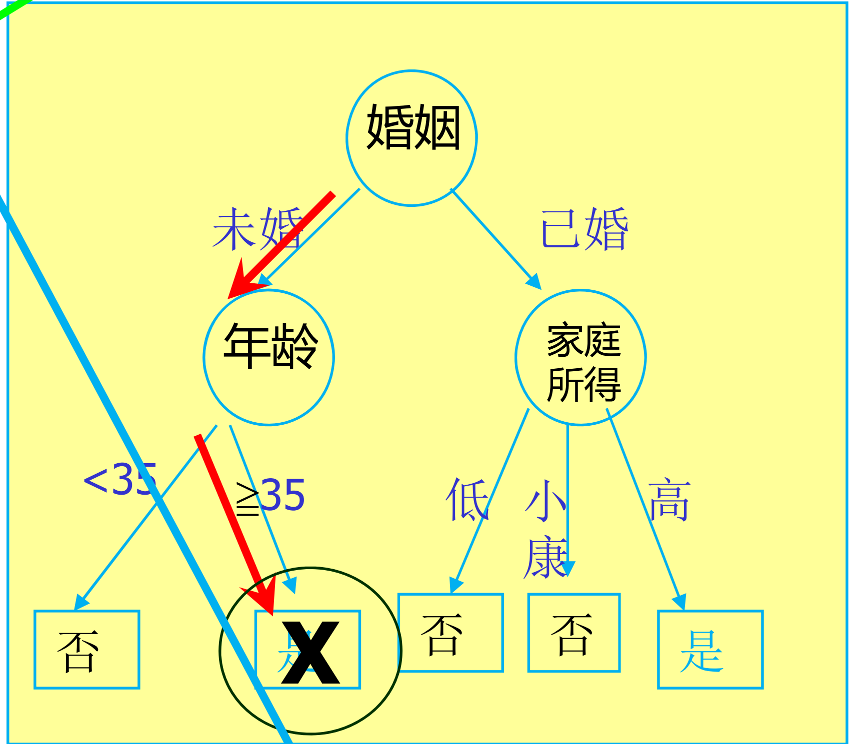
1.Build Model

Data

編號	性別	年齡	婚姻	家庭所得	購買RV房車
A0001	Male	<35	未婚	高所得	否
A0002	Male	<35	未婚	小康	否
A0003	Female	≧35	已婚	高所得	是
A0004	Male	≧35	未婚	低所得	是
A0005	Female	≧35	已婚	高所得	否
A0006	Male	≧35	已婚	低所得	否
A0007	Female	≧35	未婚	小康	否
A0008	Male	≧35	已婚	高所得	是
A0009	Male	<35	已婚	低所得	是

Training

Test



	預測結果
是	錯誤
是	正確
否	錯誤

Accuracy=33.3%

Modify Model

3.Prediction

編號	性別	年齡	婚姻	家庭所得	購買RV房車
W0144	Male	55	已婚	高所得	?

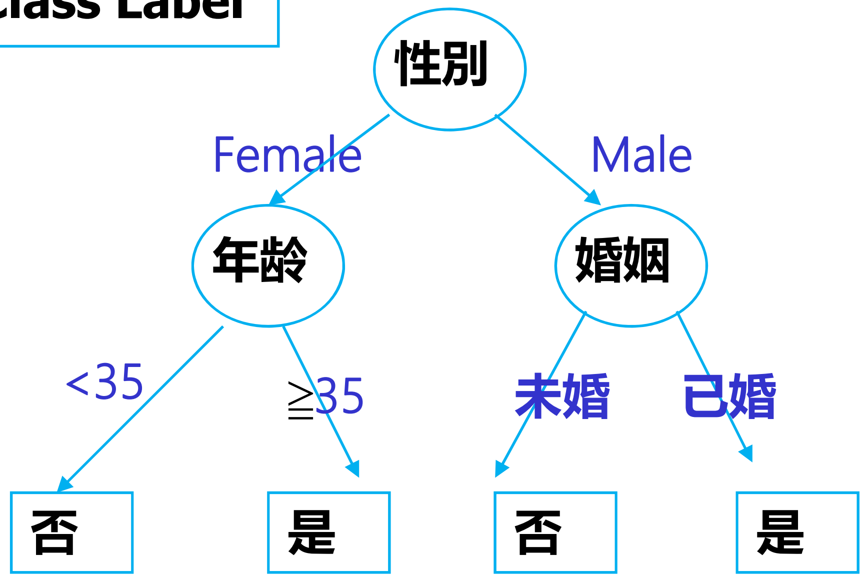
Supervised learning (classification)

Decision Tree

Data

編號	性別	年齡	婚姻	家庭人數	購買RV房車
A0001	Male	45	未婚	1	是
A0002	Male	52	已婚	7	是
A0003	Female	38	已婚	5	是
A0004	Male	25	已婚	5	否
A0005	Female	48	已婚	4	是
A0006	Male	32	未婚	3	是
A0007	Female	65	已婚	4	否
A0008	Male	33	已婚	3	是
A0009	Male	45	已婚	4	是
A0010	Female	52	未婚	1	是
A0011	Male	38	未婚	1	否
…	…	…	…	…	…
Z0099	Male	22	未婚	4	是

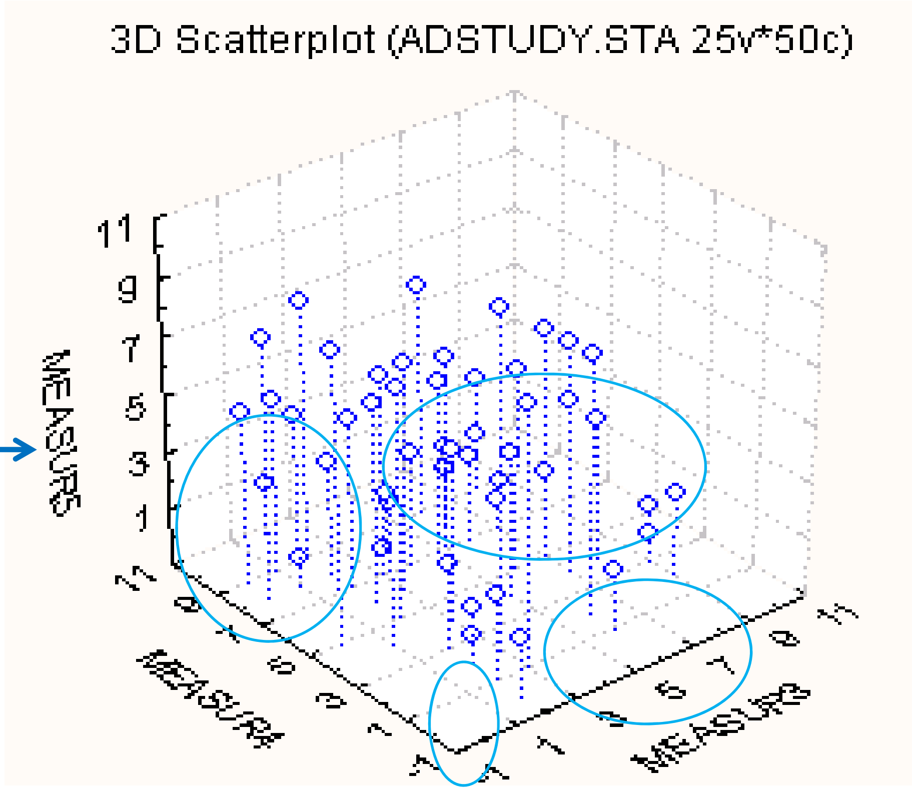
Class Label



Unsupervised learning (clustering)

Cluster Analysis

編號	性別	年齡	婚姻	家庭人數
A0001	Male	45	未婚	1
A0002	Male	52	已婚	7
A0003	Female	38	已婚	5
A0004	Male	25	已婚	5
A0005	Female	48	已婚	4
A0006	Male	32	未婚	3
A0007	Female	65	已婚	4
A0008	Male	33	已婚	3
A0009	Male	45	已婚	4
A0010	Female	52	未婚	1
A0011	Male	38	未婚	1
…	…	…	…	…
Z0099	Male	22	未婚	4



Data Preparation

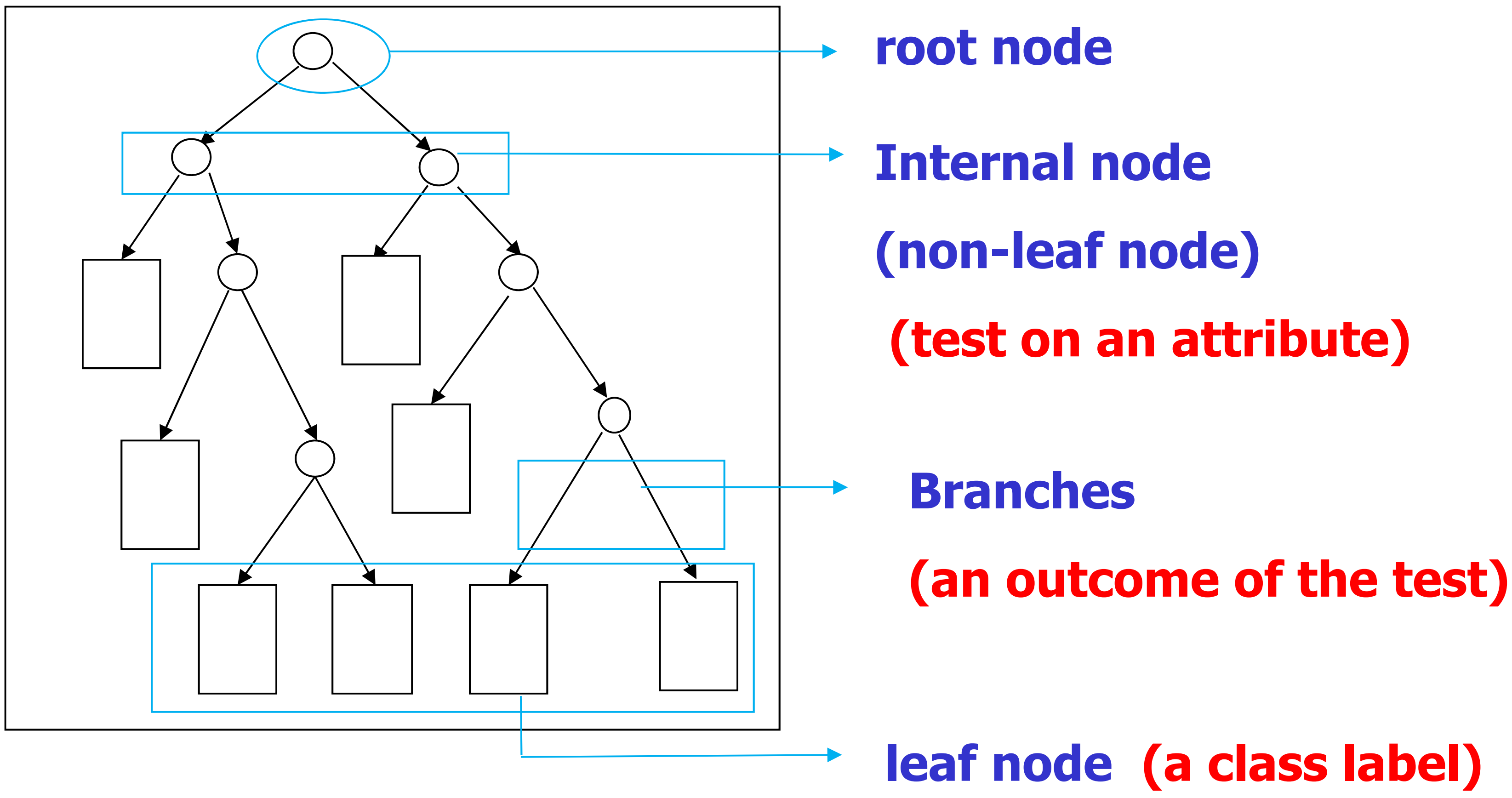
- Data cleaning
  - Preprocess data in order to reduce noise and handle missing values
- Relevance analysis (feature selection)
  - Remove the irrelevant (ID) or redundant attributes (age - birth)
- Data transformation
  - Generalize and/or normalize data
- Above discussed in Lesson 2 (If lost, pick up it.)

Lesson 7 Classification (I)

- Basic Concepts
- Decision Tree Induction
- Bayesian Classification
- Backpropagation
- Support Vector Machines (SVM)
- Lazy Learners (kNN)
- Other Classification Methods
- Additional Topics
- Prediction
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Summary

Decision Tree Induction

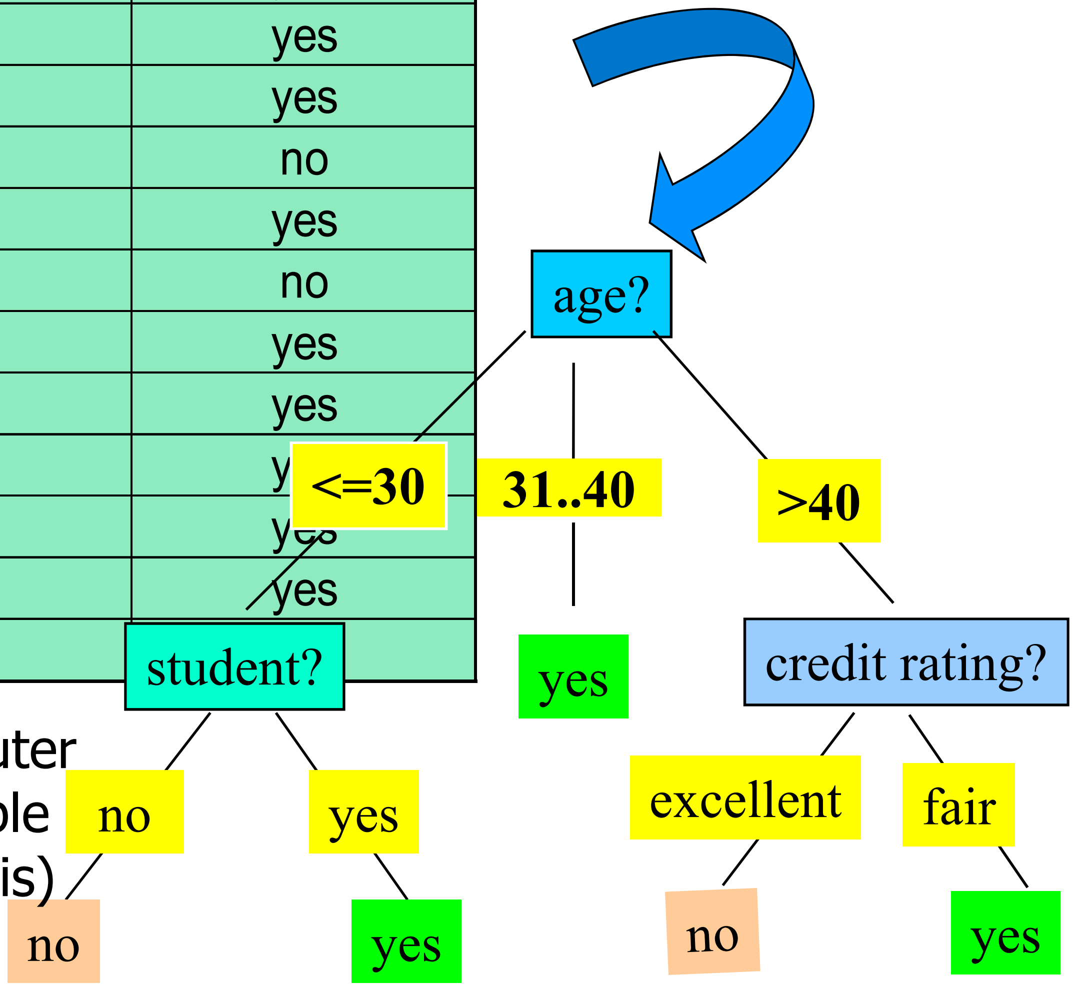
- **A flowchart-like tree structure**



Decision Tree Induction: An Example

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	y
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	[illegible]

Resulting tree:



- Training data set: buys_computer
- The data set follows an example of Quinlan's ID3 (Playing Tennis)

Algorithm for Decision Tree Induction

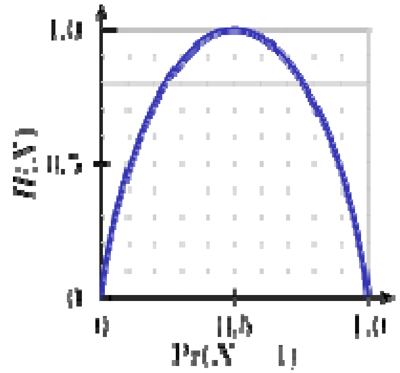
- Basic algorithm (a greedy algorithm)
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they are discretized in advance)
  - Examples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain, gain ratio, gini index)

Algorithm for Decision Tree Induction

- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left, that is, a *partition Dj is empty*

Brief Review of Entropy

- Entropy (Information Theory)
  - A measure of uncertainty associated with a random variable
  - Calculation: For a discrete random variable *Y* taking *m* distinct values $\{y_1, \dots, y_m\}$,
    - $H(Y) = -\sum_{i=1}^{m} p_i \log(p_i)$, where $p_i = P(Y = y_i)$
  - Interpretation:
    - Higher entropy => higher uncertainty
    - Lower entropy => lower uncertainty
- Conditional Entropy
  - $H(Y|X) = \sum_x p(x) H(Y|X = x)$



m = 2

Attribute Selection Measure: Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- This attribute minimizes the information needed to classify the tuples in the resulting partitions and reflects the least randomness or impurity in these partitions, i.e. best separate a given data partition.
- Let D be the training dataset, |D| is number of tuples in D
- *m* classes, C_i (i = 1, …, m). $C_{i, D}$ is the set of tuples of class C_i in D. $|C_{i, D}|$ is the number of tuples in $C_{i, D}$
- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_{i, D}|/|D|$
- Expected information (entropy) needed to classify a tuple in D:

$$Entropy(D) = Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

Attribute Selection Measure: Information Gain (ID3/C4.5)

- Suppose attribute A having *v* distinct values is used to split D into *v* partitions or subsets $\{D1, \ldots, Dv\}$. Ideally, each partition is pure.
- Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

- Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

- Minimize the amount of information ($Info_A(D)$) required to finish classifying the tuples

Computation of Information Gain for *age*

- Class P: buys_computer = "yes" (9)
- Class N: buys_computer = "no" (5)

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = 0.694$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

$\frac{5}{14}I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes'es and 3 no's. Hence

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Similarly,

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

$$Gain(credit\_rating) = 0.048$$

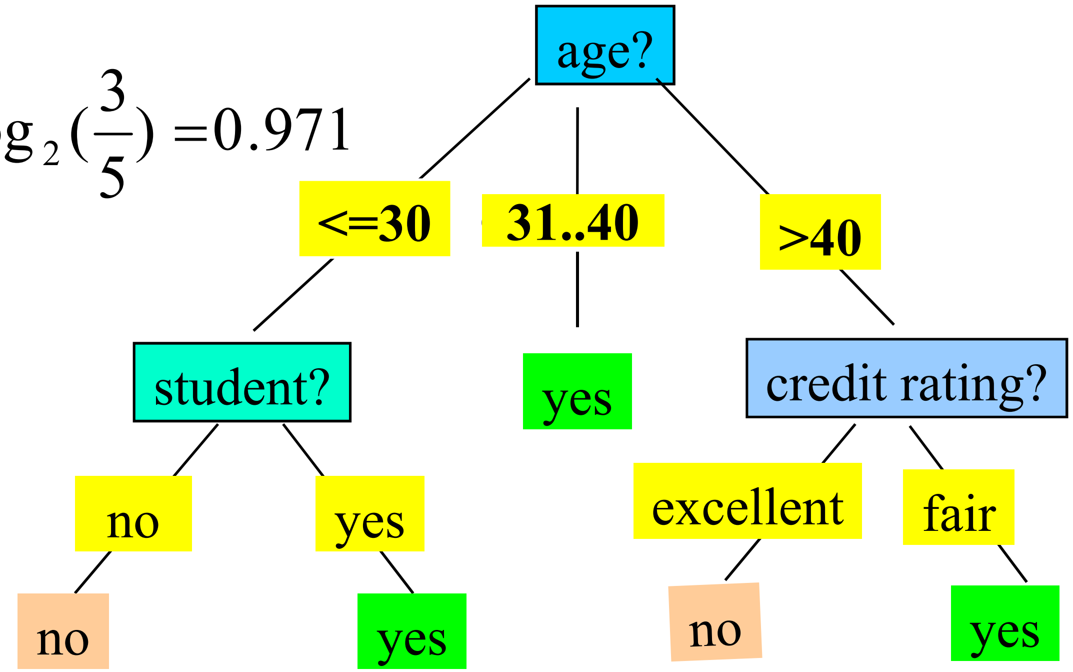
Computation of Information Gain for *age*

$$Info_{age}(D) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2)$$
$$= 0.694$$

$$I(2,3) = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right) = 0.971$$

$$I(4,0) = 0$$

$$I(3,2) = I(2,3) = 0.971$$



- So, the maximal information gain is obtained on attribute *age* and then *age* was selected as the first node for decision tree.

Computing Information-Gain for Continuous-Value Attributes

- Let attribute A be a continuous-valued attribute
- Must determine the *best split point* for A
  - Sort the value A in increasing order
  - Typically, the midpoint between each pair of adjacent values is considered as a possible *split point*
    - $\frac{a_i + a_{i+1}}{2}$ is the midpoint between the values of a_i and a_{i+1}
  - The point with the *minimum expected information requirement* for A is selected as the split-point for A
- Split:
  - D1 is the set of tuples in D satisfying A ≤ split-point, and D2 is the set of tuples in D satisfying A > split-point

Attribute Selection Measure: Gain Ratio (C4.5)

- Information gain measure is biased towards attributes with a large number of values (Cumstomer_id ? worst!)
- C4.5 (a successor of ID3) uses gain ratio to overcome the problem (normalization to information gain)

$$SplitInfo_A(D) = -\sum_{j=1}^{V} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

- Gain Ratio(A) = Gain(A) / SplitInfo(A)

Computation of Gain Ratio for *income*

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

$$Info_{income}(D) = \frac{4}{14}I(3,1) + \frac{6}{14}I(4,2) + \frac{4}{14}I(2,2) = 0.911$$

$$Gain(income) = Info(D) - Info_{income}(D) = 0.029$$

$$SplitInfo_{income}(D) = -\frac{4}{14}\times\log_2\left(\frac{4}{14}\right) - \frac{6}{14}\times\log_2\left(\frac{6}{14}\right) - \frac{4}{14}\times\log_2\left(\frac{4}{14}\right)$$

$$= 1.557$$

- gain_ratio(*income*) = 0.029/1.557 = 0.019
- The attribute with the maximum gain ratio is selected as the splitting attribute

Gini Index (CART, IBM IntelligentMiner)

- gini index is used for binary decision trees.
- gini index measures the impurity of dataset D.
- If a data set *D* contains examples from *n* classes, gini index, *gini* (*D*) is defined as

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

where p_j is the relative frequency of class *j* in *D*

- If a data set *D* is split on attribute A into two subsets D_1 and D_2, the *gini* index $gini_A(D)$ is defined as

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

Gini index (CART, IBM IntelligentMiner)

- Reduction in Impurity: (the larger the better)

$$\Delta gini(A) = gini(D) - gini_A(D)$$

- Gini index, gini (D) is fixed given D, thus, the smallest value of gini $_A$(D) , the A split is the best; that is, the largest Reduction in Impurity is the best split.
- The attribute provides the smallest $gini_{split}(D)$ (or the largest reduction in impurity) is chosen to split the node (*need to enumerate all the possible splitting points for each attribute*)
- Given one attribute A having *v* different values, the possible subsets is 2^V; After excluding the power set and the empty set, there are 2^V -2 possible ways to form two partitions of the data D based on a binary split on A.

Computation of Gini index (CART, IBM IntelligentMiner)

- Ex. D has 9 tuples in buys_computer = "yes" and 5 in "no"

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- Suppose the attribute *income* partitions D into 10 in D_1: {low, medium} and 4 in D_2 : {high}

$$gini_{income \in \{low, medium\}}(D) = \frac{10}{14} gini(D_1) + \frac{4}{14} gini(D_2)$$

$$= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right)$$

$$= 0.443$$

$$= Gini_{income \in \{high\}}(D).$$

$gini_{\{low,high\}}$ is 0.458; $gini_{\{medium,high\}}$ is 0.450. Thus, split on the {low,medium} (and {high}) since it has the lowest Gini index

- *Income* is selected as split attribute instead of *age* at the root node by gini index; while, *age* is selected by information gain for nonbinary tree.

Computation of Gini index (CART, IBM IntelligentMiner)

- All attributes are assumed continuous-valued
- May need other tools, e.g., clustering, to get the possible split values
- Can be modified for categorical attributes

Comparing Attribute Selection Measures

- The three measures, in general, return good results but
  - Information gain:
    - biased towards multivalued attributes
  - Gain ratio:
    - tends to prefer unbalanced splits in which one partition is much smaller than the others
  - Gini index:
    - biased to multivalued attributes
    - has difficulty when # of classes is large
    - tends to favor tests that result in equal-sized partitions and purity in both partitions

Other Attribute Selection Measures

- CHAID: a popular decision tree algorithm, measure based on χ^2 test for independence
- C-SEP: performs better than info. gain and gini index in certain cases
- G-statistics: has a close approximation to χ^2 distribution
- MDL (Minimal Description Length) principle (i.e., the simplest solution is preferred):
  - The best tree as the one that requires the fewest # of bits to both (1) encode the tree, and (2) encode the exceptions to the tree
- Multivariate splits (partition based on multiple variable combinations)
  - CART: finds multivariate splits based on a linear comb. of attrs.
- Which attribute selection measure is the best?
  - Most give good results, none is significantly superior than others

Overfitting and Tree Pruning

- Overfitting: An induced tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Poor accuracy for unseen samples
- Two approaches to avoid overfitting
  - Prepruning: Halt tree construction early—do not split a node if this would result in the goodness measure falling below a threshold
    - Difficult to choose an appropriate threshold
  - Postpruning: Remove branches from a "fully grown" tree—get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the "best pruned tree"

Enhancements to Basic Decision Tree Induction

- Allow for **continuous-valued attributes**
  - Dynamically define new discrete-valued attributes that partition the continuous attribute value into a discrete set of intervals
- Handle **missing attribute values**
  - Assign the most common value of the attribute
  - Assign probability to each of the possible values
- **Attribute construction**
  - Create new attributes based on existing ones that are sparsely represented
  - This reduces fragmentation, repetition, and replication

Classification in Large Databases

- Classification—a classical problem extensively studied by statisticians and machine learning researchers
- Scalability: Classifying data sets with millions of examples and hundreds of attributes with reasonable speed
- Why decision tree induction in data mining?
  - relatively faster learning speed (than other classification methods)
  - convertible to simple and easy to understand classification rules
  - can use SQL queries for accessing databases
  - comparable classification accuracy with other methods

Scalable Decision Tree Induction Methods

- SLIQ (EDBT'96 — Mehta et al.)
  - Builds an index for each attribute and only class list and the current attribute list reside in memory
- SPRINT (VLDB'96 — J. Shafer et al.)
  - Constructs an attribute list data structure
- PUBLIC (VLDB'98 — Rastogi & Shim)
  - Integrates tree splitting and tree pruning: stop growing the tree earlier
- **RainForest** (VLDB'98 — Gehrke, Ramakrishnan & Ganti)
  - Builds an AVC-list (attribute, value, class label)
- BOAT (PODS'99 — Gehrke, Ganti, Ramakrishnan & Loh)
  - Uses bootstrapping to create several small samples

Scalability Framework for RainForest

- Separates the scalability aspects from the criteria that determine the quality of the tree
- Builds an AVC-list: **AVC (Attribute, Value, Class_label)**
- **AVC-set** (of an attribute *X*)
  - Projection of training dataset onto the attribute *X* and class label where counts of individual class label are aggregated
- **AVC-group** (of a node *n*)
  - Set of AVC-sets of all predictor attributes at the node *n*

Rainforest: Training Set and Its AVC Sets

Training Examples

age	income	student	credit_rating	comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

AVC-set on *Age*

Age	Buy_Computer	
	yes	no
<=30	2	3
31..40	4	0
>40	3	2

AVC-set on *income*

income	Buy_Computer	
	yes	no
high	2	2
medium	4	2
low	3	1

AVC-set on *Student*

student	Buy_Computer	
	yes	no
yes	6	1
no	3	4

AVC-set on *credit_rating*

Credit rating	Buy_Computer	
	yes	no
fair	6	2
excellent	3	3

BOAT (Bootstrapped Optimistic Algorithm for Tree Construction)

- Use a statistical technique called *bootstrapping* to create several smaller samples (subsets), each fits in memory
- Each subset is used to create a tree, resulting in several trees
- These trees are examined and used to construct a new tree *T'*
  - It turns out that *T'* is very close to the tree that would be generated using the whole data set together
- Adv: requires only two scans of DB, an incremental alg.

Lesson 7 Classification (I)

- Basic Concepts
- Decision Tree Induction
- Bayesian Classification
- Backpropagation
- Support Vector Machines (SVM)
- Lazy Learners (kNN)
- Other Classification Methods
- Additional Topics
- Prediction
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Summary